

ПОСТРОЕНИЕ ИНТЕРАКТИВНЫХ СИСТЕМ ВИЗУАЛЬНОГО СОПРОВОЖДЕНИЯ УЧЕБНЫХ ЗАДАЧ НА ОПЕРИРОВАНИЕ ПРОСТРАНСТВЕННЫМИ ОБРАЗАМИ

Аннотация

Рассмотрены ключевые вопросы построения интерактивных систем визуального сопровождения учебных задач с пространственными образами. Представлена реализация экспериментальной системы визуального сопровождения с применением технологий захвата движений пользователя.

Ключевые слова: захват движений, пространственные образы, внеосевая проекция, интерполяция движений, стереометрия.

ВВЕДЕНИЕ

Множество учебных дисциплин ставит перед учащимися задачи, в которых требуется оперировать трехмерными объектами различной конфигурации. В первую очередь к таким дисциплинам относятся стереометрия и черчение, также подобная необходимость возникает в физике, химии, биологии и других дисциплинах. Цель обозначенных систем как можно более правильно представить учащемуся объект его исследования, кроме того, подобные системы позволяют улучшить восприятие печатного материала, наглядный переход от чертежа к объемному изображению способствует развитию пространственного мышления и установлению корректных ассоциативных связей между чертежом в учебнике и отвечающим ему мысленным пространственным образом [1].

При рассмотрении существующих систем визуального сопровождения занятий (GeomSpace, Geometria, Sabri 3D и другие) был выявлен их общий недостаток, который заключается в отставании от современных интерактивных технологий. А именно, недостаточное внедрение современных средств формирования объемных изображений и бесконтактного взаимодействия чело-

века с вычислительной машиной. В ходе исследований в указанной области была создана система Sterizium (рис. 1, верхняя часть), в процессе разработки которой был решен ряд задач, направленных на внедрение новых технологий, и рассмотрены наиболее важные моменты, связанные с построением подобного рода систем.

Предложенную систему можно разбить на три основных подсистемы (рис. 1, нижняя часть): графическая подсистема, подсистема захвата движений пользователя и подсистема ввода учебных данных.

ГРАФИЧЕСКАЯ ПОДСИСТЕМА

Визуализация графического материала может быть осуществлена средствами специализированных графических библиотек. Для построения системы Sterizium была выбрана библиотека OpenGL [2], поскольку она является кроссплатформенной.

Основной сложностью при построении изображения на современных графических картах является корректная визуализация полупрозрачных объектов. Порядок отрисовки полигонов должен начинаться с самых дальних граней. Методы сортировки с разбиением пересекающихся граней требуют больших вычислительных ресурсов и слож-

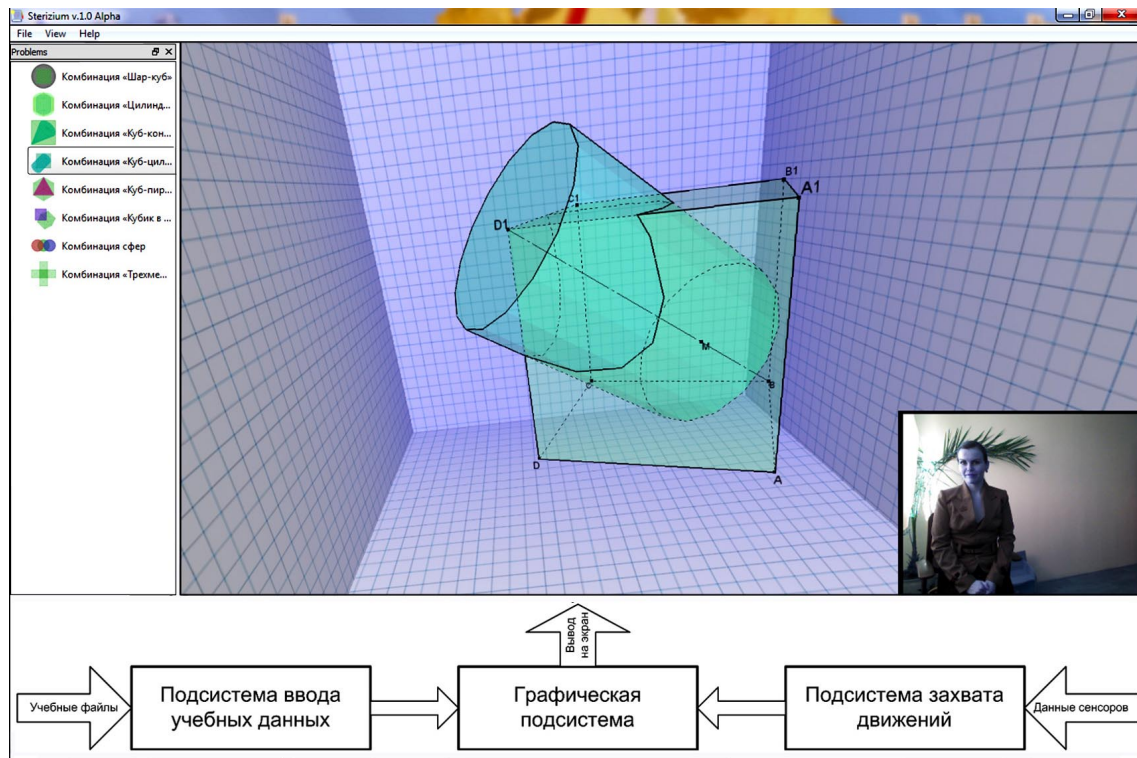


Рис. 1. Система Sterizium и основные подсистемы

ны в реализации. Наиболее современным и простым подходом является метод расслоения глубины [3], но данный метод требует многопроходной отрисовки сцены и практически не поддерживается малобюджетными видеокартами. Наиболее перспективной технологией визуализации, которая может быть применена в системах рассматриваемого класса, видится метод трассировки лучей [4], который позволяет получить фотореалистичное изображение графической сцены, что имеет большое значение для восприятия. Современное оборудование уже позволяет добиться трассировки лучей в реальном времени, но до массового применения производительность средней видеокарты должна возрасти хотя бы на порядок.

При построении системы Sterizium было принято компромиссное решение, которое обеспечивает совместимость с подавляющим большинством современных видеокарт и выглядит достаточно корректно. Суть предлагаемого метода заключается в отделении первого слоя лицевых граней отобра-

жаемых объектов от остальных лицевых граней, которые затем рисуются в задний слой в произвольном порядке. Алгоритм отрисовки выглядит следующим образом:

1. Разрешить отрисовку только лицевых граней.
2. Включить запись в буфер глубины.
3. Отключить запись в буфер цвета.
4. Отрисовать все грани в буфер глубины.
5. Отключить запись в буфер глубины.
6. Включить запись в буфер цвета.
7. Установить тест глубины, разрешающий запись пикселей, расположенных только за границей буфера глубины.
8. Отрисовать все грани.
9. Установить тест глубины, разрешающий запись пикселей, расположенных только на границе буфера глубины.
10. Отрисовать все грани.

Данный подход обеспечивает приемлемый вид при минимальных требованиях к видеокарте и ресурсам процессора.

В отношении графических проекций было выдвинуто два требования:

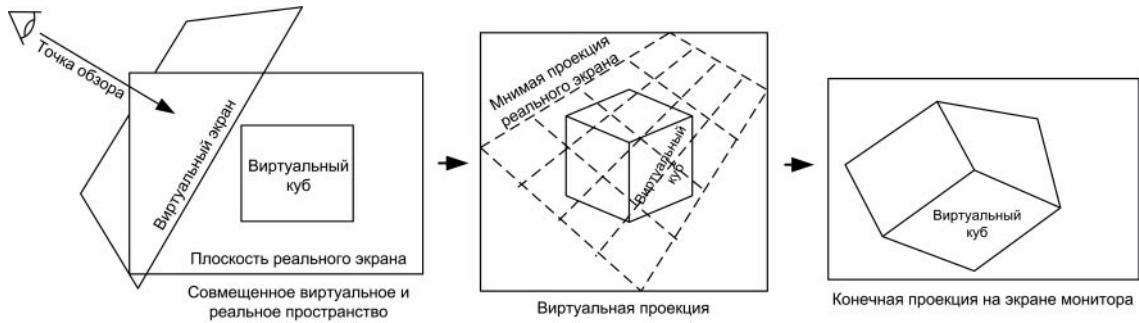


Рис. 2. Двухпроходное внеосевое проецирование

1) возможность построения корректной проекции относительно заданной позиции пользователя, для обеспечения псевдоголографического режима отображения [5];

2) возможность перехода между перспективной и ортогональной проекциями учебного графического материала с учетом общих правил отображения контуров и ребер на чертежах.

Для получения корректной относительно положения пользователя проекции на экране, как правило, используют метод внеосевого проецирования [6]. Но у данного метода есть один небольшой визуальный дефект: линии, которыми отображаются контуры и ребра моделей, будут иметь фиксированную толщину, и чем сильнее смещается пользователь от центра экрана, тем тоньше выглядит линия. Чтобы скорректировать это, был разработан двухпроходной метод построения внеосевой проекции (рис. 2).

На первом проходе осуществляется отрисовка сцены в виртуальный экран (текстуру), расположенный нормально взгляду пользователя, причем размер виртуального экрана должен быть таким, чтобы в него можно было полностью вписать проекцию реального экрана.

На втором проходе участок полученной текстуры, соответствующий мнимой проекции экрана, проецируется в буфер кадра, который увидит пользователь. Наложение в данном случае является нелинейным и аппроксимируется сеткой, число ячеек которой $N \times N$ задается формулой:

$$N = 2 \cdot \lfloor \sqrt{|\varphi|} \rfloor + 1,$$

где φ – угол между плоскостями виртуального и реального экранов в градусах. Важно отметить, что разрешение текстуры виртуального экрана не должно уступать разрешению проекции на экране пользователя.

Переход от перспективной проекции к ортогональной и обратно осуществляется посредством плавной коррекции ряда параметров (рис. 4, средний ряд). Основным параметром при этом является положение виртуальной камеры относительно плоскости проекции. Оно должно меняться от точки обзора пользователя до точки, которую условно можно принять за бесконечно удаленную точку, соответствующую ортогональной проекции (рис. 3). Когда угол α достигнет заданного минимума α_{\min} и угол β станет равным нулю, можно включать настоящее ортогональное проецирование. Среди ос-

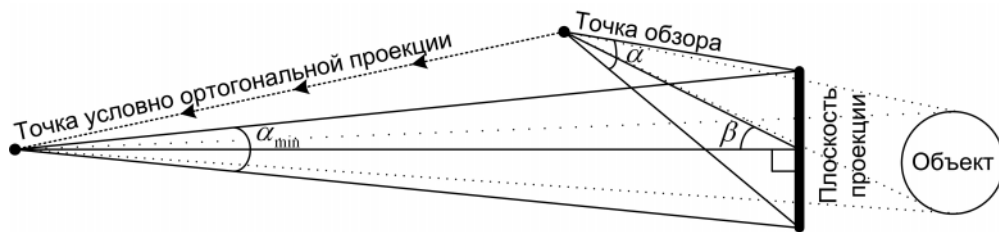


Рис. 3. Процесс смены проекции

тальных параметров стоит отметить те, которые управляют эффектами глубины и объема, их влияние должно быть сведено к минимуму при переходе в ортогональную проекцию, это закрашка полигонов, формирование освещения и тумана.

Контур объекта, включающего в себя нелинейные поверхности, такие как шар, конус или тор, определяется путем проверки граничащих полигонов на положение относительно наблюдателя. Контур пересечения объектов вычисляется один раз в момент загрузки моделей в систему, чтобы снизить объем вычислений в процессе отрисовки.

Линии контуров и ребер рисуются на заключительном этапе отрисовки. Чтобы лицевые и невидимые контуры и ребра отличались по типу линии, используется тест глубины для определения того, видна линия пользователю или скрыта от него, запись в буфер глубины при этом должна быть отключена. Чтобы лицевые линии не утопали в полигонах, их вершины необходимо приподнять в направлении нормалей к поверхности моделей.

ПОДСИСТЕМА ЗАХВАТА ДВИЖЕНИЙ

Для отслеживания действий пользователей бесконтактным методом проводится множество исследований, и уже доступен целый ряд решений, которые делятся на два класса: маркерные системы захвата движений и безмаркерные [7]. Первый класс систем не подходит, поскольку требует специальной экипировки пользователя. Второй класс систем строится на основе технологий компьютерного зрения и распознавания образов. Здесь стоит отметить молодую и активно развивающуюся библиотеку алгоритмов OpenCV [8], а также выделить два типа сенсорных систем: активные и пассивные.

Активные сенсоры выполняют сканирование пространства для получения информации о расстоянии до пользователя. В сенсорах Kinect, Xtion и DepthSense сканирование выполняется инфракрасными лучами, в пресс-релизах сенсора жестов Leap Motion говорится об использовании ультразвуковых волн. Пассивные системы используют толь-

ко видеоданные с одной и более видеокамер, что требует применения более сложных алгоритмов. Среди пассивных систем можно отметить профессиональную систему захвата движений Organic Motion, отечественную систему, ориентированную на систему недорогих Web-камер – iPi Desktop Motion Capture, но наибольший интерес представляют системы для персонального, а не хореографического применения, такие как FaceTrackingAPI от Seeing Machines. Хотя в точности пассивные системы существенно отстают от активных систем, их развитие видится более перспективным, а активные системы представляются как временная мера, позволяющая компенсировать несовершенство современных алгоритмов распознавания образов.

В качестве сенсора в системе Sterizium был использован Kinect как наиболее доступное и качественное решение на момент начала разработки (начало 2012 года). Вся работа с сенсором была вынесена в отдельный программный модуль, что в дальнейшем позволит упростить модификацию или замену подсистемы захвата движений.

Подсистема захвата движений определяет положение основных точек скелета пользователя. На основе данных о положении головы вычисляется угол обзора объекта (рис. 4, верхний ряд). А на основе данных о положении кистей рук формируются команды управления: переход от плоского чертежа к объемному изображению и обратно отталкивающим жестом левой руки в направлении экрана (рис. 4, средний ряд), а также вращение объекта правой рукой (рис. 4, нижний ряд). На сегодняшний день не существует каких-либо общепринятых, интуитивно понятных систем команд на основе жестов, данная область взаимодействия человека и машины еще слишком молода и находится на этапе своего становления.

При отслеживании положения головы пользователя на бюджетных ЭВМ был выявлен эффект прерывистой смены проекции, связанный с тем, что процессору не хватало производительности и система не успевала достаточно часто обновлять положение пользователя, хотя точность при этом не

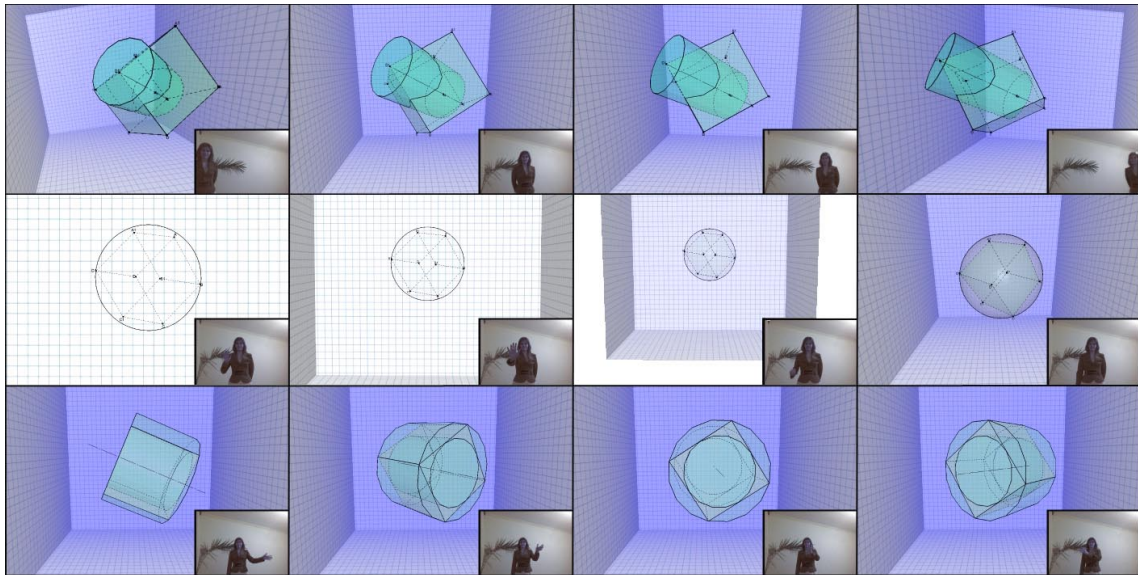


Рис. 4. Захват движений пользователя

ухудшалась. В качестве решения возникшей проблемы был предложен простой метод интерполяции движений пользователя. Интерполируемая координата p определяется формулой:

$$p = (p_i + p_{i-1} + p_{i-2}) \cdot (1 - k) + p_i \cdot k,$$

где p_i , p_{i-1} и p_{i-2} три последних точки в декартовых координатах, полученных от подсистемы захвата движений, а коэффициент k определяется следующим образом:

$$k = \begin{cases} \frac{t - t_i}{\Delta}, & \text{при } t - t_i < \Delta \\ 1, & \text{при } t - t_i \geq \Delta \end{cases}$$

здесь $\Delta = \frac{t_i - t_{i-2}}{2}$, а t_i , t_{i-2} и t — отметки

времени соответствующие рассматриваемым точкам.

Предложенный метод вносит небольшую задержку (равную периоду обновления координат), взамен обеспечивает плавную смену точки обзора экрана пользователем при недостаточной производительности подсистемы захвата движений.

ПОДСИСТЕМА ВВОДА УЧЕБНЫХ ДАННЫХ

Ввод графической информации, как правило, обеспечивается специальными редак-

торами, как внешними, так встроенными в приложение. Но в данном случае наилучшим образом подходит строгое математическое описание визуальных моделей в виде некоторого скрипта. В подавляющем большинстве задач расположение и размеры объектов задаются строго с математической точностью, и дизайнерский подход здесь неуместен.

Для ввода данных в системе Sterizium было разработано описание визуальных моделей, основанное на XML (Extensible Markup Language) [9]. Все визуальные модели должны располагаться внутри корневого тега `<root>`.

Каждая визуальная модель, предназначенная для решения конкретной педагогической задачи, описывается внутри тега `<content>`, количество моделей неограничено. Единственным и обязательным атрибутом тега содержимого является его имя *name*.

Внутри тега `<content>` допускается три типа тэгов: тег объекта `<object>`, тег осей `<axis>` и тег меток `<label>`. Каждый из указанных тэгов может быть включен в содержимое модели неограниченное число раз.

Объекты поддерживают атрибут типа *type*, посредством которого может быть задан эллипс, призма, цилиндр, конус, тор и

другие геометрические объекты с указанием их параметров через дополнительные атрибуты. Если же тип объекта не указан, то считается, что объект задан вручную в виде списка вершин **<vertexes>**, полигональных сеток **<polygones>** и отображаемых ребер **<edges>**. Список вершин задается в декартовых координатах, а полигональные сетки и ребра указываются в виде набора индексов ссылающихся на список вершин.

Объекты, включающие в себя нелинейные поверхности, такие как шар, конус, тор и прочие, поддерживают атрибут **detail**, который задает степень сегментации для увеличения детализации. Для каждого объекта может быть задан цвет полигональной сетки через атрибут **color**.

Оси **<axis>** задаются идентично вершинам **<vertexes>** в декартовых координатах, при этом каждая пара точек описывает отдельно взятую ось.

Метки **<label>** используются для символического обозначения вершин и отображения точек в заданной через атрибуты **x**, **y** и **z** координате. Текст, отображаемый рядом с точкой, задается атрибутом **text**, по умолчанию текст будет располагаться в противоположном началу координат направлении относительно заданной точки. Чтобы зафиксировать положение текста, предусмотрен атрибут **ori**.

Для управления размещением и масштабированием объектов предусмотрен тег операций **<exec>**. Если данный тег находится внутри тега объекта, тогда операции производятся только над данным объектом, если тег находится выше (внутри тега содержимого), тогда операция затрагивает все содержимое визуальной модели по окончании всех операций, касающихся отдельных объектов. Предусмотрено три типа операций: вращение **rotate**, масштабирование **scale** и перемещение **move**.

Предложенный способ описания позволяет относительно легко и быстро описывать строгие трехмерные модели (рис. 5) и вполне достаточен для большинства учебных задач.

ЗАКЛЮЧЕНИЕ

Предложенные решения были полностью реализованы в системе Sterizium и показали свою эффективность при сопровождении учебного материала в рамках стереометрии, повышая его восприятие и, что еще важнее, активно привлекая внимание учащихся к поставленной задаче, вызывая живую заинтересованность.

Ознакомиться с экспериментальной версией приложения и подробными инструкциями по его использованию можно по адресу в интернете: <http://www.sterizium.ru>.

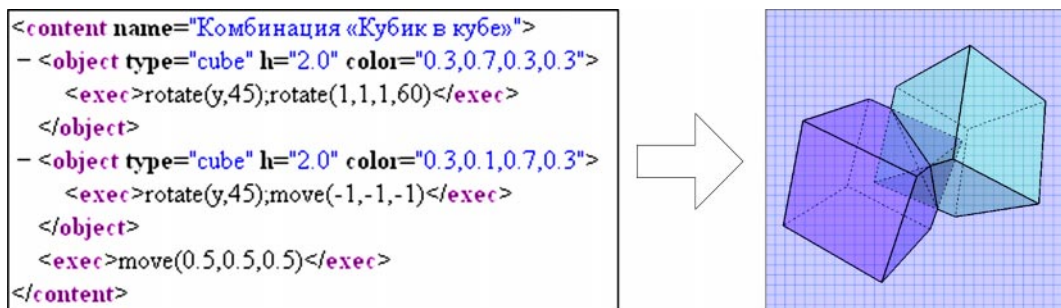


Рис. 5. Пример описания визуальной модели

Литература

1. Санина Е.И., Гришина О.А. Роль задач в формировании пространственного мышления в процессе обучения геометрии учащихся старших классов / Сборник трудов Международной научной конференции «Геометрия и геометрическое образование в современной средней и высшей школе» (к 70-летию В.А. Гусева), 22–25 ноября 2012 / Тольятти: Издательство ТГУ, 2012. С. 157–159.

2. Херн Д., Бейкер М.П. Компьютерная графика и стандарт OpenGL – 3-е изд. М.: Вильямс, 2005.
3. Bavoil L., Myers K. Order Independent Transparency with Dual Depth Peeling – Nvidia, Feb. 2008 [Electronic resource]. URL: http://developer.download.nvidia.com/SDK/10/opengl/src/dual_depth_peeling/doc/DualDepthPeeling.pdf (дата обращения 07.01.2013).
4. Боголепов Д.К., Трушанин В., Турлатов В.Е. Интерактивная трассировка лучей на графическом процессоре // Труды международной конференции по компьютерной графике и зрению «Графи-Кон'2009», 5–9 октября 2009, Москва, Россия. С. 263–266
5. Гришина О.А. Формирование пространственных образов стереометрических комбинаций с использованием новых информационных технологий // Мир Науки, Культуры, Образования, 2012. № 6(37). С. 146–148.
6. Uprur B., Sahiner A. V., Fidaner I.B. Off-Axis Stereo Projection and Head Tracking for a Horizontal Display – ACE'09, October 29–31, 2009, Athens, Greece [Electronic resource]. URL: <http://www.cmpe.boun.edu.tr/medialab/horizdisplay.pdf> (дата обращения 05.08.2012).
7. Grest D. Marker-Free Human Motion Capture – LAP Lambert Acad. Publ., 2010.
8. Bradsky G., Kaehler A. Learning OpenCV – O'Reilly, 2008.
9. Extensible Markup Language (XML) 1.1 (Second Edition) [Electronic resource]. URL: <http://www.w3.org/TR/xml11/> (дата обращения 17.10.2012).

Abstract

Discussed key issues of building interactive system for visual maintenance of the training tasks with spatial images. Shown an implementation of the experimental system for visual maintenance using motion capture technology.

Keywords: motion capture, spatial images, off-axis projection, motion interpolation, solid geometry.



Наши авторы, 2013.
Our authors, 2013.

*Гришина Ольга Анатольевна,
аспирантка РУДН (г. Москва),
olga@sterizium.ru*